AFRL-RI-RS-TR-2008-197
Final Technical Report
July 2008

DESIGN OF EFFICIENT SYNCHRONIZATION PROTOCOLS FOR WIRELESS AIRBORNE NETWORKS

Texas A & M University

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

NOTICE AND SIGNATURE PAGE

AFRL-RI-RS-TR-2008-197 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/s/	/s/
GREGORY HADYNSKI Work Unit Manager	WARREN H. DEBANY, JR Technical Advisor Information Grid Division

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
JUL 2008

2. REPORT TYPE
Final

3. DATES COVERED *(From - To)*
MAR 07 – JAN 08

4. TITLE AND SUBTITLE

DESIGN OF EFFICIENT SYNCHRONIZATION PROTOCOLS FOR WIRELESS AIRBORNE NETWORKS

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA8750-07-1-0061

5c. PROGRAM ELEMENT NUMBER
62702F

6. AUTHOR(S)

Erchin Serpedin

5d. PROJECT NUMBER
CITE

5e. TASK NUMBER
TE

5f. WORK UNIT NUMBER
XS

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Texas A & M University
1470 William d Fitch PKWY
College Station, TX 77845-4645

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFRL/RIGC
525 Brooks Rd
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-RI-RS-TR-2008-197

12. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. PA# WPAFB 08-08-3811

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The goal of this project was to develop efficient clock synchronization schemes to ensure robust operation of wireless airborne networks in the absence of GPS (Global Positioning Systems), and in the presence of mobility, time-varying channel conditions, and ad-hoc network topologies. The major contributions of this project include the development of analytical performance tools to assess the performance of clock synchronization protocols and development of novel clock synchronization protocols to achieve the ultimate performance limits predicted by the analytical performance benchmarks. The work conducted in this project lead also to the development of two novel time synchronization protocols: Adaptive Synchronization Protocol and Pairwise Broadcast Synchronization (PBS) Protocol, respectively, which were shown to exhibit a number of desirable features (such as scalability, adaptivity, energy-efficient and distributed capabilities) for clock synchronization in wireless airborne networks and wireless sensor networks.

15. SUBJECT TERMS
Wireless airborne networks, network protocols, time synchronization

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
U	U	U	UU	51	Gregory Hadynski
					19b. TELEPHONE NUMBER (*Include area code*) 315-330-4094

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

Acknowledgements

The PI would like to express his heartfelt thanks to Mr. Gregory Hadynski and Dr. Bruce Suter from Air Force Research Laboratory (AFRL), Rome, NY, for all their support, help and encouragement.

1 Summary

The main goal of this project was to develop efficient clock synchronization schemes to ensure robust operation of wireless airborne networks in conditions of mobility, time-varying channel conditions, ad-hoc decentralized infrastructures and absence of GPS (Global Positioning Systems). To achieve this goal, a number of tasks were accomplished in this project. The first task dealt with in this project was the development of analytical performance benchmarks to assess the performance of clock synchronization protocols for wireless ad-hoc networks at micro-level, i.e., sub-networks involving the synchronization of two or a limited number of adjacent nodes. This first task was fully addressed in this project, and efficient analytical tools to assess the performance of clock synchronization algorithms in terms of the Cramer-Rao lower bound (CRLB) and mean-square error (MSE) were developed. The second task of this project focused on designing efficient and robust clock synchronization algorithms that achieve these ultimate performance limits, and on validating the performance of these algorithms through computer simulations. The second task was also successfully addressed in this project. Finally, the third major task of this project was the design of energy-efficient protocols for synchronization of large-scale wireless ad-hoc airborne networks. Two novel protocols entitled Adaptive-Sync Protocol and PBS (Pairwise Broadcast Synchronization) Protocol, respectively, were developed for synchronization of large-scale wireless airborne networks and shown to exhibit all the desirable features: scalability, distributivity, adaptivity, energy-efficiency, decentralized operation, and ability to perform local as well as global synchronization. The work conducted in this project was reported in numerous journal and conferences, and in several PhD and MSc Thesis Dissertations at Texas A&M University that are publicly available. Overall, this project lead to a large number of novel results of paramount importance in the area of clock synchronization of wireless ad-hoc networks and more

general networks such as Internet. Some of these results could be viewed as real breakthroughs. As examples of such breakthrough results, this project developed clock synchronization schemes that achieve the ultimate performance limits, outperform significantly the existing state-of-the-art schemes and are robust to the distribution of network delays. High-performance extensions of the most representative protocols RBS (Reference Broadcast Synchronization) and TPSN (Timing Synch Protocol for Sensor Networks) proposed for clock synchronization of wireless ad-hoc (sensor) networks were also derived. The proposed PBS and Adaptive-Sync protocols represent high-performance energy-efficient extensions of RBS and TPSN protocols, respectively.

2 Introduction

Future wireless airborne networks are envisioned to represent the next frontier of networking, to be pervasive and ubiquitous, and to provide a wide range of services and applications. This trend is underlined by a number of technological advances and demands. The rapidly growing demands for mobility and anywhere-anytime data access represents a major driving force behind the next generation of mobile wireless airborne networks. Recent technological developments mark also the departure of telecommunications systems from homogeneous networks to heterogeneous networks, from non-intelligent devices to smart devices, and from telephony-based services to multi-media services. In addition, recent advances in hardware and inexpensive wireless radio systems have made also possible the design of low-cost, low-power, and multi-functional sensor devices. When deployed in a large number across a geographical area, these sensor devices create a self-organized cooperative ad-hoc network that is perfectly fit for distributed sensing and automated information gathering, processing and communication.

Wireless sensor networks have been recognized as a revolutionary technology that will have a huge impact on a broad range of applications: monitoring the health status of humans and environment, control and instrumentation of industrial machines and home appliances, energy-conservation, security, detection of chemical and biological leaks, etc. Wireless sensor networks are a special case of wireless ad hoc networks, and assume a multi-hop communication framework with no infrastructure and where the sensors cooperate spontaneously by forwarding each other's packets from a source to a destination node. The upcoming years will very likely witness a growing demand for more intelligent sensor systems that will be networked with wireless local area networks (WLANs), Internet, satellite and Unmanned Aerial Vehicle (UAV) networks to create a global wireless airborne network with increased functionality and performance.

In general, for distributed computing and networking systems, maintaining the logical clocks of the computers in such a way that they are never too far apart is one of the most complex problems of computer engineering. Whether it is the disciplining of computer clocks with the devices synchronized to a Global Positioning System (GPS) satellite or a Network Time Protocol (NTP) time server over the Internet, it is possible to equip some primary time servers for the purpose of synchronizing a much larger number of secondary servers and clients connected through a common infrastructure. In order to do this, a distributed network clock synchronization protocol is required through which a server clock can be read, the readings to other clients can be transmitted and each client clock can be adjusted as required. In such a distributed synchronization approach, the participating devices exchange timing information with their chosen reference at regular intervals and adjust their logical clocks accordingly.

A computer clock in general has two components, namely a frequency source and a means of accumulating timing events (consisting of a clock interrupt mechanism and a counter implemented in software). The implementation of the computer clock in the operating system and the programming interface differ between operating systems and hardware platforms. However, the basic source of timing are an uncompensated quartz crystal oscillator and the clock interrupts it generates. Theoretically, two clocks would remain synchronized if their offsets are set equal and their frequency sources run at the same rate. However, practical clocks are set with limited precision and the frequency sources run at slightly different rates. In addition, the frequency of a crystal oscillator varies due to initial manufacturing tolerance, aging, temperature, pressure and other factors. Because of these inherent instabilities, distributed clocks must regularly be synchronized to keep them running close to each other.

The Network Time Protocol [23] represents the most widely used clock synchronization protocol for large-scale networks with static topology such as the Internet. In NTP, the nodes are externally synchronized to a global reference time that is represented in the network by a set of master nodes or time servers that are referred to as layer-1 servers. The entire synchronization process assumes a hierarchical tree organization of the network nodes. Despite its wide-spread use in the synchronization of Internet, NTP is not appropriate for synchronization of wireless ad-hoc sensor networks that are subject to severe energy-constraints, dynamic topologies caused by mobility and node failures, and absence of GPS and global time references (due to either jamming, interferences, or absence of direct line of sight communication links). In addition, the service provided by NTP assumes continuous synchronization of all the network nodes with maximum accuracy and with no concern about energy consumption. However, NTP is not equipped with an mechanism to enable the local synchronization of a subset of nodes, and to keep the rest of the nodes switched to a power-saving (sleeping) state. Since listening continuously for the synchronization beacons is an energy-consuming operation, NTP can not directly be applied to synchronization of energy-constrained wireless ad-hoc networks as is the case with wireless airborne networks and wireless sensor networks.

These considerations illustrate the need for novel distributed and scalable synchronization protocols for wireless ad-hoc networks that in general must satisfy a series of requirements: energy-efficiency, robustness with respect to node mobility and link/node failures, and ability to guarantee the long-term network synchronization at local and global scales.

Clock synchronization is important for many applications such as Internet delay measurements, cellular networks, data security algorithms, Media Access Control (MAC) protocols like Time Division Multiple Access (TDMA), Internet Protocol (IP) telephony, ordering of updates in database systems, etc. Recently, with the advent of Wireless Sensor Networks (WSNs) and Wireless Airborne Networks (WANs), developing clock synchronization protocols that suit their specific requirements is becoming an important research problem [30]. A large number of their applications require the clocks of the nodes to run synchronously on a common timescale. This is the case with applications such as data fusion, efficient duty cycling operations, acoustic beamforming, localization, security and object tracking. Unlike conventional networks, energy efficiency must also be taken into account for addressing the clock synchronization problem in WSNs and WANs.

3 Methods, Assumptions and Procedures

During the last two decades, many clock synchronization protocols have been proposed such as [4], [5], [23], etc. NTP [23] is a protocol for synchronizing the clocks of computer systems over packet-switched, variable-latency data networks and it represents the Internet standard for time synchronization. It is a layered client-server architecture based on the User Data Protocol (UDP) message passing which synchronizes computer clocks in a hierarchical way using the offset delay estimation method. NTP's sender-receiver synchronization architecture is widely accepted in designing time synchronization algorithms and consists of the same two-way timing message exchange mechanism targeted in this paper.

A protocol based on the remote clock reading method was put forward by [5], which handles unbounded message delays between processes. In [4], the time transmission protocol is used by a node to communicate the time on its clock to a target node, which subsequently estimates the time in the source node by using message timestamps and message delay statistics.

For ad-hoc communication networks, the time synchronization protocol [28] represented one of the pioneering contributions in this area. The protocol is based on generating timestamps to record the time at which an event of interest occurred. The timestamps are updated by each node using its local clock and the time transformation method, where the final timestamp is expressed in terms of an interval with a lower bound and an upper bound.

In the realm of wireless sensor networks, the clock synchronization protocols of particular note are Reference Broadcast Synchronization (RBS [7]), Timing Synch Protocol for Sensor Networks (TPSN [8]) and Time Diffusion Protocol (TDP [21]). RBS relies on simultaneous reception of broadcast pulses by several nodes transmitted by a common neighboring node after which the nodes exchange their timestamps and estimate the relative time offsets and skews. On the other hand, TPSN is based on the same sender-receiver paradigm as in NTP, like many other traditional clock synchronization protocols. The basic difference is that TPSN has been molded sufficiently to suit the requirements of wireless sensor networks. On the other side, TDP establishes a network-wide equilibrium time through an iterative, weighted averaging technique based on a diffusion of messages involving all the nodes in the synchronization process.

In general, there are two different approaches for synchronizing a pair of nodes which can be categorized as sender-receiver synchronization (SRS) and receiver-receiver synchronization (RRS). The former is based on the classical model of two-way message exchanges between a pair of nodes. In contrast, the latter compares the time readings of a beacon packet from a common sender at a set of nodes. Most of the existing time synchronization protocols rely on one of these two approaches. For instance, RBS is based on RRS since it requires pairs of message exchanges among children nodes (except the reference) to compensate their relative clock offsets, while TPSN adopts SRS since it depends on a series of pairwise synchronizations that assume two-way timing message exchanges.

The work conducted in this project focused in the following main directions:

- development of a comprehensive body of analytical results to assess the ultimate performance limits achievable by clock synchronization protocols, and design of energy-efficient protocols to achieve these limits.

- development of energy efficient protocols for global synchronization of wireless ad-hoc (sensor) networks.
- design of synchronization schemes that are robust to the high-latencies induced by large propagation delays, clock skew variations, and network traffic delays.

This research work plays a fundamental role in the design of future generation of energy-efficient long-lived wireless airborne networks. The studies offered by current protocols (RBS [7], TPSN [8], Flooding Time Synchronization Protocol (FTSP) [22], tiny/mini synchronization [32], diffusion algorithm [21], etc.) have concentrated on short-term and small-scale sensor network synchronization, and are not fit for synchronization of long-term and large-scale networks that require efficient duty-cycling mechanisms in the presence of continuous time synchronization (e.g., coordinated actuation and synchronized sampling) and/or sensor-initiated (post-facto) synchronization. The lack of techniques to accurately predict the performance of time synchronization protocols has been recently recognized as an important obstacle in designing long-lived wireless sensor networks.

By eliminating the empiric and highly suboptimal design principles used in the current synchronization protocols, the design principles established in this project will help reduce dramatically the re-synchronization rate, reduce the signaling rate and transmission packet overhead, optimize the operation of existing synchronization protocols, and help in establishing the foundations for the design of future generation of long-lived energy-efficient sensor networks. One of the major benefits of the work conducted in this project is the development of the energy-efficient Adaptive-Sync and PBS (Pairwise Broadcast Synchronization) protocols for long-term synchronization of sensor networks. The Adaptive-Sync Protocol exhibits a number of very attractive features for synchronization of sensor networks:

- it represents a significantly enhanced and natural extension of NTP and TPSN for global synchronization of large-scale ad-hoc networks
- it aims at minimizing the energy consumption in large-scale and long-lived sensor networks
- it is equipped with flexible mechanisms to adjust the synchronization mode (local versus global), (re-) synchronization period, and clock phase-offset and skew estimators in order to achieve long term reliability
- it employs sequential message exchange techniques and energy-efficient signaling schemes to reduce further the number of RF-transmissions
- as opposed to RBS, TPSN, and FTSP protocols that perform poorly in high-latency networks, the Adaptive-Sync Protocol is also fit for networking environments characterized by high propagation delays and clock skew variations, and exhibits robustness to unknown network delay distributions.

4 Results and Discussion

In the present section, we will discuss the major results that were accomplished in this project.

4.1 Efficient Clock-Offset Compensation Schemes for Local Network Synchronization

The very first step in designing an efficient synchronization protocol is to develop efficient schemes for pairwise synchronization of two nodes, and to understand what are the ultimate performance limits and to design estimation schemes that achieve these limits. From the theory of point estimation [19], it is known that once an analytical model is available for a given estimation method, then under some general conditions, a lower bound for the variance of any unbiased estimator can be expressed in terms of the Cramer-Rao lower bound (CRLB). Furthermore, techniques to build estimators that achieve the ultimate performance predicted by Cramer-Rao lower bound are available (e.g., the maximum likelihood estimation principle, the Rao-Blackwell and Lehmann-Scheffe constructions of minimum variance unbiased estimators) [19], [15]. These were the basic ideas that this project followed to assess the ultimate performance of existing synchronization schemes and to develop synchronization schemes that achieve these performance limits.

The common synchronization protocols NTP and TPSN rely on the clock offset correction scheme between two nodes that assumes the pairwise message exchange mechanism depicted in Fig. 1. Since the NTP and TPSN protocols assumes no clock skew (frequency offset) between the two nodes, we will improve by a significant margin the performance of NTP and TPSN protocols by the introduction of a clock skew correction mechanism within the same message exchange protocol and without causing any additional overhead. First we will derive the maximum likelihood estimator (MLE) and the Cramer-Rao lower bound for the clock phase-offset in the presence of Gaussian network delays and no clock skew, and then address the issue of clock skew compensation.

Assume that the message exchanges that take place between two generic nodes A and B are the ones depicted in Fig. 1-a. Node A sends its time reading $T_{1,i}$ to Node B, which records its time of arrival $T_{2,i}$ according to its own timescale. A similar timing message exchange is sent from Node B to Node A. The ith delay observations corresponding to the ith timing message exchange are given by $U_i = d + \theta_A + X_i$ and $V_i = d - \theta_A + Y_i$, respectively (using similar notations as in [1]), and are graphically represented in Fig. 1-a. The fixed value θ_A denotes the clock offset between the two nodes, X_i and Y_i denote the variable portions of delays which are assumed to be normal distributed random variables (RVs) with mean μ and variance σ^2, respectively.

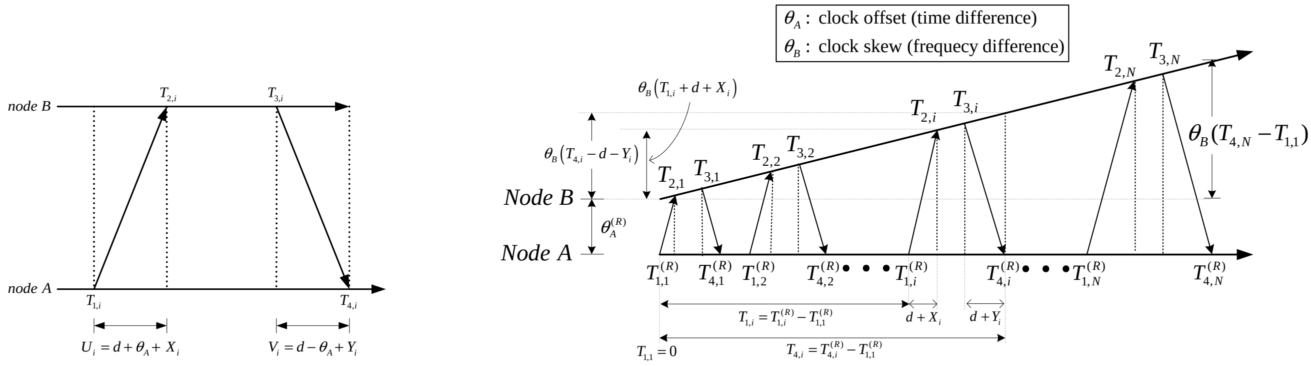

Figure 1: a) Message exchange between master-slave nodes having clock offset. b) Message exchange between master-slave nodes having clock offset and skew.

Maximizing the likelihood function based on the observations $\{X_i\}_{i=1}^N$ and $\{Y_i\}_{i=1}^N$, we arrive at the MLE of clock offset:

$$\hat{\theta}_A = \arg\max_{\theta_A} [\ln L(\theta_A)] = \frac{\sum_{i=1}^{N} (U_i - V_i)}{2N} = \frac{\overline{U} - \overline{V}}{2}, \tag{1}$$

where $\overline{U}$ and $\overline{V}$ stand for the sample means of observations $\{U_i\}_{i=1}^N$ and $\{V_i\}_{i=1}^N$, respectively. The CRLB of estimator (1) takes the form:

$$\operatorname{var}(\hat{\theta}_A) \geq CRLB(\hat{\theta}_A) = \frac{\sigma^2}{2N} \ . \tag{2}$$

The computer simulations illustrate that the mean-square error (MSE) of MLE (1) is well predicted by the CRLB (2) for any number of observations N (see Fig. 2-a). It is interesting to observe that the expression and performance of MLE strongly depends on the type of distribution assumed by network delays (see Fig. 2-b). We have shown (see the references [27], [31]) that when the network delays are exponentially distributed with mean α, the MLE and corresponding

CRLB for the clock phase offset assume the expressions:

$$\hat{\theta}_A = \frac{\min\limits_{1\le i\le N} U_i - \min\limits_{1\le i\le N} V_i}{2}, \tag{3}$$

$$\mathrm{var}(\hat{\theta}_A) \ge CRLB(\hat{\theta}_A) = \frac{\alpha^2}{4N^2}. \tag{4}$$

Extensive computer simulations illustrate that the performance of MLEs (1) and (3) is not robust with respect to (wrt) the network delay distribution. In other words, the Gaussian MLE (1) and exponential MLE (3) perform very poorly in the presence of exponential and Gaussian network delays, respectively. Notice that this disproportionate behavior is corroborated by the totally different dependence of CRLBs (2) and (4) with respect to the number of observations N (linear vs. quadratic). This suggests the need for robust estimation schemes that exhibit good performance irrespective of the underlying distribution of network delays. Our preliminary computer simulation results (see our preliminary publication [18]) show that the clock-offset estimators built within the framework of sequential Markov-Chain Monte-Carlo techniques are robust with respect to the distribution of network delays and present excellent performance.

Next, based on the same pairwise message exchange mechanism, we will develop a novel approach to compensate both the clock phase offset as well as the skew. Since every oscillator has its unique clock frequency, the clock offset between two nodes generally keeps drifting away. Also, since a fixed value model for clock offset is not sufficient, estimating the difference of clock frequencies (i.e., clock skews) between two nodes increases the synchronization accuracy and guarantees long-term reliability without frequent re-synchronization. We will next illustrate the main challenges behind determining efficient joint ML-estimators for clock phase and skew offset, and then show a technique to build high-performance estimators for clock-phase and skew offset that are robust to high-latencies (unknown possible large propagation delays). Due to

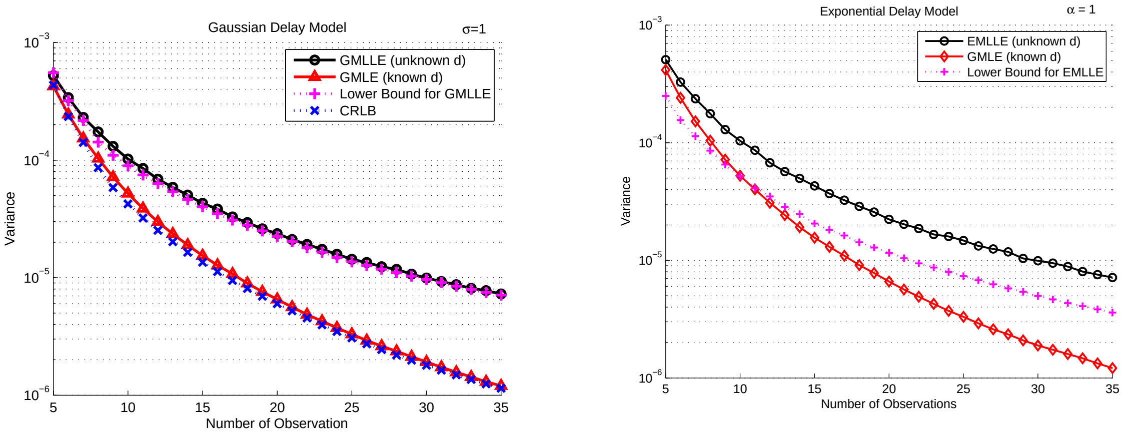

Figure 2: a) Variance of MLE for the Gaussian delay model and the proposed estimator for Gaussian random delays ($\sigma^2 = 1$). b) Variance of MLE for the Gaussian delay model and the proposed estimator for exponential random delays ($\alpha = 1$).

space limitations, we will only sketch the main results. Fig. 1-b shows the effect of clock skew (denoted by θ_B) on the timing message exchange between the two nodes A and B. In this model, the ith received signal at the node B, $T_{2,i}$, is given by

$$T_{2,i} = T_{1,i} + d + X_i + \theta_B \left(T_{1,i} + d + X_i\right) + \theta_A = \left(1 + \theta_B\right)\left(T_{1,i} + d + X_i\right) + \theta_A, \qquad (5)$$

where the additional term $\theta_B \left(T_{1,i} + d + X_i\right)$ is due to the effect of clock skew. Similarly, the ith received signal at the node A, $T_{4,i}$, is expressed as

$$T_{4,i} = T_{3,i} + d + Y_i - \theta_B \left(T_{4,i} - d - Y_i\right) - \theta_A, \qquad (6)$$

$$T_{3,i} = \left(1 + \theta_B\right)\left(T_{4,i} - d - Y_i\right) + \theta_A, \qquad (7)$$

where the term $\theta_B \left(T_{4,i} - d - Y_i\right)$ is again due to the drift caused by clock skew, and (7) is obtained by re-arranging (6). Assuming that $\{X_i\}_{i=1}^{N}$ and $\{Y_i\}_{i=1}^{N}$ are zero mean independent Gaussian distributed RVs with variance σ^2, and the fixed portion of delay d is known, the Maximum Likelihood (ML) estimates for θ_A and θ_B can be expressed in closed-form as a function of unknown delay parameter d (a result derived in more details in our publication [27]). Notice further that in general framework networks that rely on Radio Frequency (RF) transmissions over short distances, d can be assumed known ($d = 0$), and the proposed MLEs of clock phase-offset and skew are directly implementable.

Since the clock difference between two wireless terminals is monotonically increasing or decreasing due to the linear clock skew model, the clock difference will be maximized between the first and last time stamps. From this intuition, novel and practical clock skew estimators can be developed by using the first and last observations of timing message exchanges. The idea is to build clock skew estimates that maximize the likelihood function based on the first and last observations of timing stamps. From (5), subtracting $T_{2,1}$ from $T_{2,N}$ gives

$$T_{2,N} - T_{2,1} = T_{1,N} - T_{1,1} + X_N - X_1 + \theta_B \left(T_{1,N} - T_{1,1} + X_N - X_1\right). \tag{8}$$

Similarly from (6), subtracting $T_{4,1}$ from $T_{4,N}$ yields

$$T_{4,N} - T_{4,1} = T_{3,N} - T_{3,1} + Y_N - Y_1 - \theta_B \left(T_{4,N} - T_{4,1} - (Y_N - Y_1)\right). \tag{9}$$

Define the differences of the first and last time stamps as $D_{(1)} = \sum_{i=2}^{N} D_{1,i} = T_{1,N} - T_{1,1}$, $D_{(2)} = \sum_{i=2}^{N} D_{2,i} = T_{2,N} - T_{2,1}$, $D_{(3)} = \sum_{i=2}^{N} D_{3,i} = T_{3,N} - T_{3,1}$, and $D_{(4)} = \sum_{i=2}^{N} D_{4,i} = T_{4,N} - T_{4,1}$, respectively, and where $D_{k,l} = T_{k,l} - T_{k,l-1}$ captures the difference between two consecutive time stamps ($k = 1, \ldots, 4$, $l = 2, \ldots, N$). Then, (8) and (9) can be rewritten respectively as

$$D_{(2)} = D_{(1)} + P + \theta_B \left(D_{(1)} + P\right),$$

$$D_{(4)} = D_{(3)} + R + \theta_B \left(D_{(4)} - R\right),$$

where $P = X_N - X_1$ and $R = Y_N - Y_1$. Since X_N, X_1, Y_N, and Y_1 are i.i.d. (independent and identically distributed) normal distributed RVs with variance σ^2, P and R are zero-mean normal distributed RVs with variance $2\sigma^2$, respectively. So the joint Probability Density Function (PDF) of P and R is given by

$$f_{P,R}(p,r) = \left(\frac{1}{4\pi\sigma^2}\right)^2 e^{-\frac{1}{4\sigma^2}\left(p^2+r^2\right)}.$$

The likelihood function with respect to $\theta'_B = 1/(1+\theta_B)$ and σ^2 takes the form

$$L\left(\theta'_B, \sigma^2\right) = \left(\frac{1}{4\pi\sigma^2}\right)^2 e^{-\frac{1}{4\sigma^2}\left[D^2_{(2)}\left(\theta'_B-\beta\right)^2+D^2_{(3)}\left(\theta'_B-\gamma\right)^2\right]}, \tag{10}$$

and by maximizing (10) with respect to θ'_B leads to the clock-skew estimator:

$$\hat{\theta}_B = \frac{1}{\hat{\theta}'_B} - 1 = \frac{D^2_{(2)} + D^2_{(3)}}{D_{(1)}D_{(2)} + D_{(3)}D_{(4)}} - 1. \tag{11}$$

This analysis can be further extended to derive efficient clock-skew estimators in the case of exponentially distributed latencies. Notice further that the proposed clock-skew estimator (11) can be used to determine efficient clock phase-offset estimators. Using (5) and (6), the ith observations of timing message exchange delays $U_i = T_{2,i} - T_{1,i}$ and $V_i = T_{4,i} - T_{3,i}$ are expressed, respectively, as

$$U_i = d + X_i + \theta_B\left(T_{1,i} + d + X_i\right) + \theta_A,$$

$$V_i = d + Y_i - \theta_B\left(T_{4,i} - d - Y_i\right) - \theta_A.$$

Since $T_{2,i}$ and $T_{4,i}$ are known values and θ_B can be estimated (11), the sets of delay observations between the two nodes can be recomposed as

$$U'_i = U_i - \hat{\theta}_B T_{1,i} \quad \left(= d' + \theta_A + X'_i\right), \tag{12}$$

$$V'_i = V_i + \hat{\theta}_B T_{4,i} \quad \left(= d' - \theta_A + Y'_i\right), \tag{13}$$

where $X'_i = (1+\theta_B)X_i$, $Y'_i = (1+\theta_B)Y_i$, and $d' = (1+\theta_B)d$, respectively. Since the clock-skew is compensated, eqs. (12) and (13) suggest that the same ML clock offset estimators as in (1) and (3) for Gaussian and exponential delay models, respectively, can be applied to arrive at the estimates:

$$\hat{\theta}_A = \frac{\overline{U_i'} - \overline{V_i'}}{2} \qquad \text{(Gaussian delays)}, \tag{14}$$

$$\hat{\theta}_A = \frac{\min\limits_{1 \le i \le N} U_i' - \min\limits_{1 \le i \le N} V_i'}{2} \qquad \text{(exponential delays)}. \tag{15}$$

Fig. 2-a compares the MSEs of the proposed clock-skew estimator (11) for the Gaussian delay model with MLE (that assumes knowledge of d) and corresponding CRLB in Gaussian random delay channels ($\sigma^2 = 1$). It can be seen that the proposed estimator (11) performs close to MLE, is consistent and asymptotically efficient. Fig. 2-b shows the variance of the counterpart estimator (11) for the exponential delay model and that of MLE (that assumes knowledge of d) for the Gaussian delay model in exponential random delay channels ($\alpha = 1$). It can also be seen that the proposed estimator is consistent and exhibits comparable performance to the ideal MLE.

4.2 Adaptive-Sync Protocol for Global Network Synchronization

The work conducted in this project also lead to the development of a significantly enhanced extension (named Adaptive-Sync Protocol) of NTP and TPSN protocols. Adaptive-Sync adaptively optimizes some crucial network parameters, such as the synchronization mode, the re-synchronization period and the number of beacons per pairwise synchronization, with respect to the current network status. The Adaptive-Sync Protocol consists of three functional phases:

- *Level discovery phase*: This phase is the same as that in TPSN, and is used for generating a hierarchical structure in the network.

- *Synchronization phase*: It is similar to the corresponding synchronization phase in TPSN. However, as opposed to TPSN, the Adaptive-Sync Protocol adjusts not only the current clock offset but also the clock skew to guarantee the long term synchronization, while TPSN only estimates the clock offset. Hence, the Adaptive Sync Protocol requires far less frequent re-synchronization.

- *Network evaluation phase*: The reference node investigates the current status of network traffic in order to select the synchronization mode between *always on* (AO) (always maintain network-wide synchronization) and *sensor initiated* (SI) (synchronize only when it needs to). Besides, it optimizes the re-synchronization period and the number of beacons per each pairwise synchronization.

The second and third phases (i.e., synchronization and the network evaluation phases) will be periodically repeated in order to minimize the overall energy consumption with respect to the current network status. The Adaptive-Sync Protocol assumes also a number of additional parameters such as latency factor, average number of hops, and timing synchronization period to optimize the overall performance of synchronization protocol. Relative to TPSN protocol, Adaptive-Sync assumes the additional *network evaluation phase*, while the functions of the other two phases present some common similarities with the ones encountered in TPSN. In the synchronization phase, Adaptive-Sync estimates not only the current clock drift but also the clock frequency (skew) to guarantee long term reliability of synchronization while TPSN only estimates the clock offset, and therefore TPSN might require more frequent network-wide synchronization. Robustness to high-latencies and distribution of network delays is ensured based on the clock estimators presented in the previous section. Furthermore, Adaptive-Sync adapts the clock offset and skew estimators in order to achieve long term synchronization.

As TPSN, generating a hierarchical structure in the network, the level discovery phase, is the first step of Adaptive-Sync. In this phase, every single node in the network will be assigned a level and gets ready for synchronization. Upon creating the network hierarchy, the reference node investigates the current status of network traffic in order to adjust the period of synchronization and select the synchronization mode between *always on* (AO - always maintain network-wide synchronization) and *sensor initiated* (SI - synchronize only when it needs to). This step stands for the network evaluation phase, and its goal is to minimize the number of message exchanges for synchronization in a given time interval, i.e., it aims to minimize total energy consumption for synchronization. The third step of Adaptive-Sync, called network-wide time synchronization, consists in the pairwise synchronization between adjacent nodes until every node in the network is synchronized to the reference(s). Adaptive-Sync periodically repeats the network evaluation and synchronization phases to minimize the total energy consumption in the network with respect to the current network status.

Another critical problem is to determine the required number of timing message exchanges per each pairwise synchronization. To achieve higher synchronization accuracies, a larger number of message transfers and an additional signal processing workload are needed during each pairwise synchronization. As the number of required timing messages per each pairwise synchronization increases, the overall number of timing messages in a synchronization period increases. Hence, there is a tradeoff between accuracy and energy consumption. To address these design challenges, next we propose an adaptive clock synchronization algorithm to aid in selecting the synchronization mode between AO and SI, the period of synchronization τ, and the number of message exchanges per pairwise synchronization N aiming at efficient usage of network resources for reduced energy consumption. The network parameters are summarized as follows:

- B: number of branches in a spanning tree of the network.
- τ: period of clock synchronization.

- $\overline{h}$: average number of hops per unit time
- δ: latency factor reflecting the amount of allowed delay in data transmission
- N: number of message exchanges per pairwise synchronization

The number of branches in the network B can be obtained after the level discovery phase. The latency factor δ should be fixed according to the type of sensor network and its range is from 0 to 1. The higher latency factor means higher concern for network delays. In every sensing event, the destination node adds the number of hops that have occurred in that particular transmission to its storage. During the next synchronization phase, the reference node collects the information of the total number of hops occurred in the last sync period and determines the average number of hops per unit time ($\overline{h}$) in the network. As mentioned earlier, the goal of the adaptive clock synchronization algorithm is to minimize the number of required timing messages. In an AO-mode, the number of timing messages per unit time is given by $\overline{M} = 2BN/\tau$, while in the SI-mode, $\overline{M} = 2\overline{h}N$. To minimize the number of timing messages per unit time $\overline{M}$, the synchronization mode should be selected as follows:

$$\frac{2BN\delta}{\tau} \underset{SI}{\overset{AO}{\lessgtr}} 2\overline{h}N, \tag{16}$$

where the latency factor δ varies from 0 to 1 such that the more delay-dependent networks assume a larger value of δ and vice versa ($0 \leq \delta \leq 1$). For example, δ is set to 0 for sensor networks requiring network synchronization all the time (only AO mode is available). On the other extreme, for delay-independent networks, δ should be close to 1.

As the period of timing synchronization τ is getting larger, the network is getting power efficient. Thus, τ should be chosen as large as possible. However, a too large value for τ induces a critical synchronization problem since the clock drift between nodes keeps generally increasing with time. Hence, there exists a maximum value of timing sync period ($\tau = \tau_{max}$)

which is determined by the oscillator characteristics (hardware specifications) and the accuracy of estimators. Notice that (16) can be rewritten as

$$\tau_{max} \underset{SI}{\overset{AO}{\gtrless}} \frac{B\delta}{\overline{h}} \ (= T_h). \tag{17}$$

Thus, the sync mode changes from AO into SI when T_h is greater than τ_{max}. In the SI-mode, the reference node periodically asks the number of hops that occurred during the past time interval, and then make a decision whether or not to switch to the AO-mode. In fact, τ_{max} is dependent on N since it strongly depends on the accuracy of timing offset estimators. A more detailed analysis of τ_{max} is provided next.

The synchronization phase performs pairwise synchronization among a set of nodes by exchanging timing messages. For the AO-mode, a series of pairwise synchronization will take place until every node in the network is synchronized to the reference node, i.e., the message exchanges are occurring at all branches of the network spanning tree. On the other hand, for the SI mode, only nodes participating in the particular multi-hop data transmission synchronize with each other. The number of timing messages per pairwise synchronization is a critical parameter to determine both the synchronization accuracy and power efficiency. The clock error (difference) ε between two nodes A and B that exchange N timing messages is modeled as follows: $\varepsilon = \varepsilon_o + \varepsilon_s t$, where t denotes the time, ε_o and ε_s stand for the clock offset and skew estimation errors, respectively. Let $\varepsilon_{o,i}$ and $\varepsilon_{s,i}$ denote the clock offset and skew estimation errors when $N = i$ message exchanges occur between the two nodes. Both these errors can be well modeled by normal distributions: $\varepsilon_{o,i} \sim N(0, \sigma^2_{\varepsilon_{o,i}})$, and $\varepsilon_{s,i} \sim N(0, \sigma^2_{\varepsilon_{s,i}})$. Therefore, $\varepsilon \sim N(0, \sigma^2_{\varepsilon})$, where $\sigma^2_{\varepsilon} = \sigma^2_{\varepsilon_{o,N}} + \sigma^2_{\varepsilon_{s,N}} \tau^2_{\max}$, $(t = \tau_{max})$. Imposing the upper-limit $\varepsilon_{\max}$ for the clock error via the probabilistic measure:

$$P\left(|\varepsilon| \leq \varepsilon_{\max}\right) = 2\text{erf}\left(\frac{\varepsilon_{\max}}{\sigma_{\varepsilon}}\right) \ , \tag{18}$$

leads to the maximum period for clock synchronization:

$$\tau_{\max}^{(N)} = \sqrt{\frac{\sigma_{\varepsilon}^2 - \sigma_{\varepsilon_{o,N}}^2}{\sigma_{\varepsilon_{s,N}}^2}}\ . \tag{19}$$

Indeed, Imposing the upper-limit $\varepsilon_{\max}$ for the clock error via the probabilistic measure:

$$P_s = Pr\left(|\varepsilon| \geq \varepsilon_{\max}\right) = erfc\left(\frac{\varepsilon_{\max}}{\sqrt{2}\sigma_{\varepsilon}}\right),$$

where $erfc(x) \triangleq 2/\sqrt{\pi} \cdot \int_x^{\infty} \exp\left(-t^2\right)dt$ and P_s denotes the synchronization error probability for pairwise synchronization. Thus, σ_{ε} can be determined when $\varepsilon_{\max}$ and the maximum allowable P_s are fixed. For instance, when P_s is limited to 0.1% and $\varepsilon_{\max}$ is 10ms, then the standard deviation of clock mismatch (σ_{ε}) has to be smaller than 3.04ms.

The maximum re-synchronization period with N beacons can be written as

$$\tau_{\max}^{(N)} = \sqrt{\frac{\sigma_{\varepsilon}^2 - \sigma_{\varepsilon_{o,N}}^2}{\sigma_{\varepsilon_{s,N}}^2}}\ . \tag{20}$$

Based on the lower bounds and asymptotic performance of the estimators, one can easily infer closed-form expressions of the variances $\varepsilon_{o,N}$ and $\varepsilon_{s,N}$ in terms of the variances $\varepsilon_{o,1}$ and $\varepsilon_{s,2}$, respectively. From the lower bounds that we derived in [27], $\sigma_{\varepsilon_{o,N}}^2$ can be written with respect to N and $\sigma_{\varepsilon_{o,1}}^2$ as

$$\sigma_{\varepsilon_{o,N}}^2 = \frac{\sigma_{\varepsilon_{o,1}}^2}{N}.$$

Similarly, since the time differences between beacons are proportional to N and by far greater than the variance of delays, the following relationship can be obtained from the lower bound for the clock skew estimator derived in [27]:

$$\sigma_{\varepsilon_{s,N}}^2 = \frac{\sigma_{\varepsilon_{s,2}}^2}{(N-1)^2}, \quad N \geq 2.$$

Therefore, for $N \geq 2$, $\tau_{\max}^{(N)}$ can be rewritten as

$$\tau_{\max}^{(N)} = (N-1)\sqrt{\frac{\sigma_{\varepsilon}^2 - \frac{\sigma_{\varepsilon_{o,1}}^2}{N}}{\sigma_{\varepsilon_{s,2}}^2}}, \qquad N \geq 2. \tag{21}$$

Note that $\varepsilon_{s,1}$ can be obtained by the specifications of the crystal oscillator, and $\varepsilon_{o,1}$ and $\varepsilon_{s,2}$ can be determined by simple experimental tests. Therefore, the maximum re-synchronization period is proportional to the number of beacons, and performing clock skew estimation will significantly increase $\tau_{\max}^{(N)}$ since $\sigma_{\varepsilon_{s,1}} \gg \sigma_{\varepsilon_{s,2}}$.

4.2.1 Number of Beacons Required for Each Pairwise Synchronization

The goal of the Adaptive-Sync Protocol is to minimize the average number of message exchanges ($\overline{M}$). Hence, from (20), finding the optimal number of beacons (N) resume to solving the following optimization problem

$$\hat{N} = \arg\min_{N} \overline{M}, \tag{22}$$

with

$$\overline{M} = \frac{2BN}{\tau_{sync}^{(N)} + \tau_{\max}^{(N)}} = \begin{cases} \dfrac{2B}{\tau_{sync}^{(1)} + \sqrt{\frac{\sigma_{\bar{\varepsilon}}^2 - \sigma_{\varepsilon_{o,1}}^2}{\sigma_{\varepsilon_{s,1}}^2}}} & N = 1 \\ \dfrac{2B}{\frac{\tau_{sync}^{(N)}}{N} + \frac{N-1}{N}\sqrt{\frac{\sigma_{\bar{\varepsilon}}^2 - \frac{\sigma_{\varepsilon_{o,1}}^2}{N}}{\sigma_{\varepsilon_{s,2}}^2}}} & N \geq 2 \end{cases},$$

where $\tau_{sync}^{(N)}$ denotes the synchronization time with N beacons and will be estimated at the reference node for different Ns when the network is first established. Once N is estimated from (22), $\tau_{\max}^{(N)}$ can be obtained from (21).

Simulation results in Fig. 3 (see [25] and [26] for more detailed results) show that the Adaptive-Sync Protocol requires a far less number of timing messages than TPSN and RBS when there exist multiple number of beacon transmissions. Moreover, the gap between the average number of required timing messages between the Adaptive-Sync Protocol and TPSN and RBS significantly increases as N increases, and thus Adaptive-Sync is by far more energy-efficient than TPSN and RBS for large Ns. Moreover, the Adaptive Synch Protocol is applicable to various different types of sensor network applications.

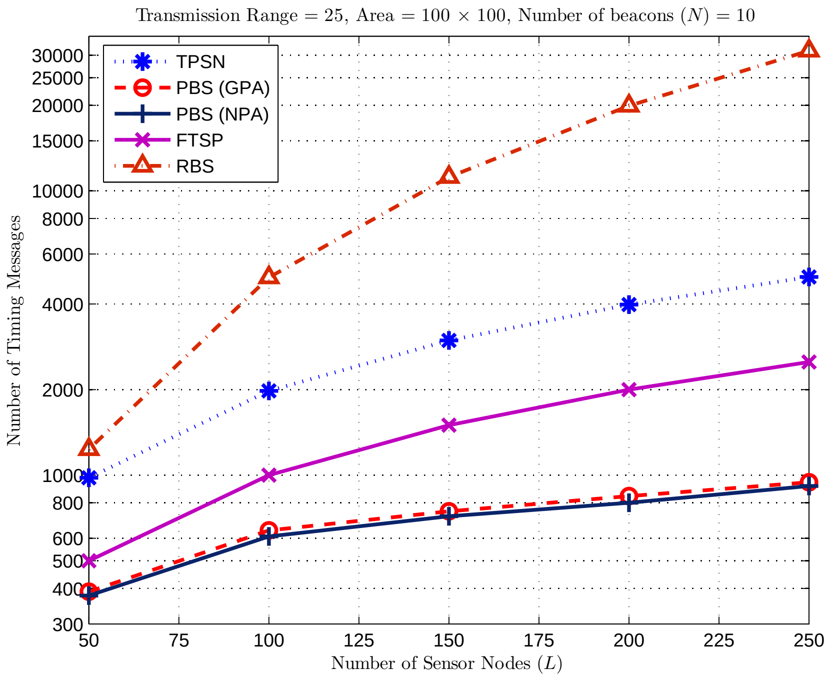

Figure 3: Required number of message exchanges with respect to the number of sensor nodes.

4.3 Pairwise Broadcast Synchronization (PBS) Protocol

A major outcome of the work conducted in this project was the development of an energy-efficient clock synchronization scheme for wireless sensor networks (WSNs) based on a novel receiver-receiver synchronization approach. Within this synchronization approach, a subset of sensor nodes are synchronized by overhearing the timing message exchanges of a pair of sensor nodes. This represents a very practical and realistic scenario for wireless airborne networks because a group of sensor nodes can be synchronized without sending any extra messages. This novel synchronization protocol brings two main contributions. The first contribution is the development of a novel synchronization approach which can be partially or fully applied for implementation of new synchronization protocols and for improving the performance of existing time synchronization protocols. The second contribution consists in the design of a time synchronization scheme which significantly reduces the overall network-wide energy consumption without incurring any

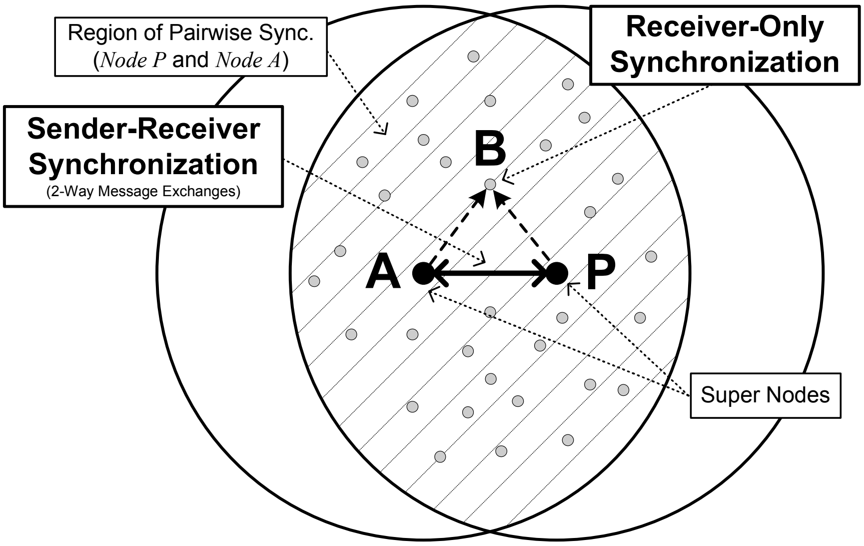

Figure 4: Receiver-only synchronization.

loss of synchronization accuracy compared to other well-known schemes.

Due to the power constraint, the communication range of a sensor is strictly limited to a (radio-geometrical) circle whose radius depends on the transmission power (see Fig. 4). In Fig. 4, every node within the checked area (e.g., *Node B*) can receive messages from both *Node P* and *Node A*. Suppose that *Node P* is a parent (or reference) node, and *Node P* and *Node A* perform a pairwise synchronization using two-way timing message exchanges. Then, all the nodes in the checked region can receive a series of synchronization messages containing the information about the time stamps of the pairwise synchronization. Here, we assume perfect communications (no data loss and failure) at the physical layer. Using this information, *Node B* can be also synchronized to the parent node *Node P* by applying a similar method as in RBS and with no extra timing messages. Indeed, *Node P* and *Node A* can be regarded as super nodes since they provide synchronization beacons for all the nodes located in their vicinity.

The work conducted in this project lead to the development of a new clock synchronization approach, named receiver-only synchronization (ROS). Similarly to *Node B* in Fig. 4, a group of sensor nodes can be synchronized by only receiving timing messages of a pairwise synchronization based on ROS. The proposed PBS scheme efficiently combines both SRS and ROS approaches to achieve network-wide synchronization with a significantly reduced number of timing messages. Next we will describe and analyze the features of the proposed synchronization scheme in more detail.

4.3.1 Pairwise Sender-Receiver Synchronization

This subsection illustrates how a parent node *Node P* and *Node A* can be synchronized using SRS. The clock model for the two-way message exchange is depicted in Fig. 5, where $\theta_{offset}^{(\mathrm{AP})}$ denotes the clock offset between *Node A* and *Node P* and timing messages are assumed to be exchanged multiple (N) times [34], [8]. Hence, the number of observations (sets of time stamps) becomes N. Here, the time stamps transmitted during the ith message exchange $T_{1,i}^{(\mathrm{A})}$ and $T_{4,i}^{(\mathrm{A})}$ are measured by the local clock of *Node A*, and $T_{2,i}^{(\mathrm{P})}$ and $T_{3,i}^{(\mathrm{P})}$ are measured by the local clock of *Node P*, respectively. *Node A* transmits a synchronization packet to *Node P*, which contains the level and identifier (ID) of *Node A* and the value of time stamp $T_{1,i}^{(\mathrm{A})}$. *Node P* receives it at $T_{2,i}^{(\mathrm{P})}$ and transmits an acknowledgement packet to *Node A* at $T_{3,i}^{(\mathrm{P})}$. This packet contains the level and ID of *Node P* and the value of time stamps $T_{1,i}^{(\mathrm{A})}$, $T_{2,i}^{(\mathrm{P})}$, and $T_{3,i}^{(\mathrm{P})}$. Then, *Node A* finally receives the packet at $T_{4,i}^{(\mathrm{A})}$.

Packet delays can be characterized into several distinct components: send, access, transmission, propagation, and receive time (see e.g., [34]). These delay components can be further divided into two parts: the fixed portions of delays in up- and down-link ($d^{(\mathrm{AP})}$, $d^{(\mathrm{PA})}$) and the variable portions of delays in up- and down-link ($X_i^{(\mathrm{AP})}$, $X_i^{(\mathrm{PA})}$), respectively.

Thus far, several random delay models have been proposed. A single-server M/M/1 queue can fittingly represent the cumulative link delay for point-to-point connection, where the random

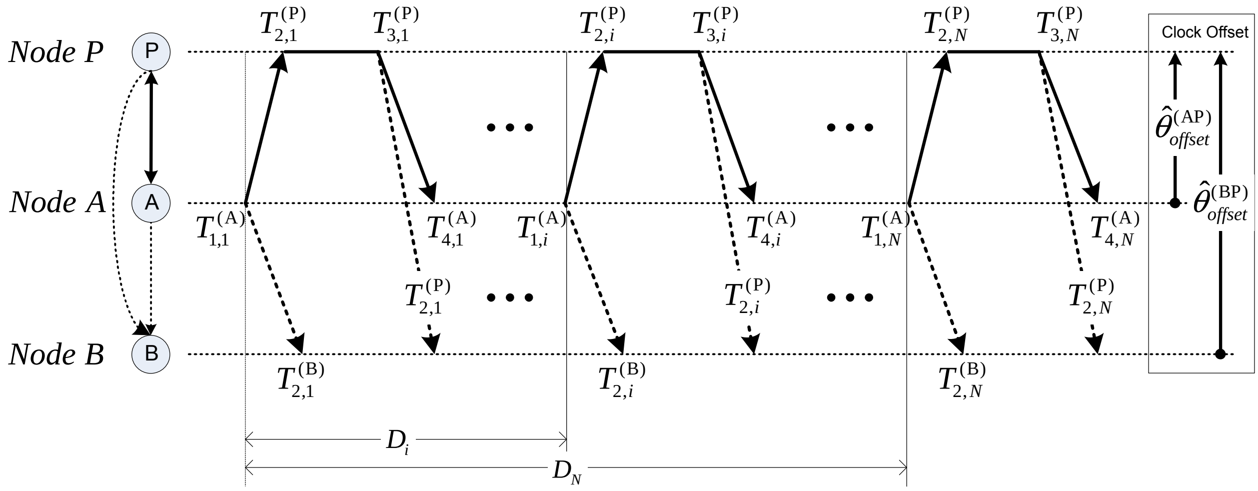

Figure 5: Clock synchronization model of PBS.

delays are modeled as exponential random variables [1]. The Gaussian delay model is appropriate if the delays are thought to be the addition of numerous independent random processes. In [7], the chi-squared test showed that the variable portion of delays can be modeled as Gaussian distributed random variables with 99.8% confidence. In this letter, $X_i^{(AP)}$ and $X_i^{(PA)}$ are assumed to be normal distributed with mean μ and variance $\sigma^2/2$.

From Fig. 5, $T_{2,i}^{(P)}$ and $T_{4,i}^{(A)}$ can be expressed as

$$\begin{aligned} T_{2,i}^{(P)} &= T_{1,i}^{(A)} + \theta_{offset}^{(AP)} + d^{(AP)} + X_i^{(AP)}, \\ T_{4,i}^{(A)} &= T_{3,i}^{(P)} + \theta_{offset}^{(PA)} + d^{(PA)} + X_i^{(PA)}, \end{aligned}$$

where $\theta_{offset}^{(PA)} = -\theta_{offset}^{(AP)}$, and $d^{(AP)}$ and $X_i^{(AP)}$ denote the fixed and random portions of timing delays in the message transmissions from *Node A* to *Node P*, respectively. In [27], the MLE of clock offset was found to be given by

$$\hat{\theta}_{offset}^{(AP)} = \arg \max_{\theta_{offset}^{(AP)}} [\ln L(\theta_{offset}^{(AP)})] = \frac{\overline{U} - \overline{V}}{2}, \quad (23)$$

with the delays in up-link $U_i \triangleq T_{2,i}^{(P)} - T_{1,i}^{(A)}$ and down-link $V_i \triangleq T_{4,i}^{(A)} - T_{3,i}^{(P)}$. From (23), *Node A* can be synchronized to the parent node *Node P* by simply taking the difference of the average delay observations $\overline{U} = 1/N \sum_{i=1}^{N} [T_{2,i}^{(P)} - T_{1,i}^{(A)}]$ and $\overline{V} = 1/N \sum_{i=1}^{N} [T_{4,i}^{(A)} - T_{3,i}^{(P)}]$. Note that applying a clock skew correction mechanism guarantees the long-term stability of synchronization, i.e., a decrease of the re-synchronization frequency. In [27], the joint maximum likelihood estimator of clock offset and skew for normal delays was derived. Although the effects of clock skew have not been considered herein, the clock skew estimators developed in [27] can be directly applied to the proposed PBS protocol with no modifications.

4.3.2 Receiver-Only Synchronization

In Fig. 4, consider an arbitrary node, say *Node B*, in the checked region. While *Node P* and *Node A* exchange time messages, *Node B* is capable of receiving packets from both nodes. Hence, *Node B* can observe a set of time readings ($\{T_{2,i}^{(B)}\}_{i=1}^{N}$) at its local clock when it receives packets from *Node A* as depicted in Fig 5. Besides, the information about a set of time stamps $\{T_{2,i}^{(P)}\}_{i=1}^{N}$ can also be obtained by receiving packets from *Node P*. Considering the effects of both clock offset and skew in this subsection, the time stamp at *Node P* in the ith uplink message $T_{2,i}^{(P)}$ is given by

$$T_{2,i}^{(P)} = T_{1,i}^{(A)} + \theta_{offset}^{(AP)} + \theta_{skew}^{(AP)} \cdot (T_{1,i}^{(A)} - T_{1,1}^{(A)}) + d^{(AP)} + X_i^{(AP)}, \tag{24}$$

where $\theta_{skew}^{(AP)}$ stands for the relative clock skew between *Node A* and *Node P*. Likewise, the time stamp at *Node B* in the ith uplink message $T_{2,i}^{(B)}$ can be represented by

$$T_{2,i}^{(B)} = T_{1,i}^{(A)} + \theta_{offset}^{(AB)} + \theta_{skew}^{(AB)} \cdot (T_{1,i}^{(A)} - T_{1,1}^{(A)}) + d^{(AB)} + X_i^{(AB)}, \tag{25}$$

where $\theta_{offset}^{(AB)}$ and $\theta_{skew}^{(AB)}$ stand for the relative clock offset and skew between *Node A* and *Node B*, $d^{(AB)}$ and $X_i^{(AB)}$ denote the fixed and random portions of timing delays in the message transmission from *Node A* to *Node B*, respectively. Here, $X_i^{(AB)}$ is assumed to be a normal distributed RV with mean μ and variance $\sigma^2/2$.

Similar to [10], [13], and [36], the linear regression technique (line fitting) can be applied to synchronize *Node B* and compensate the effects of the relative clock skew between *Node P* and *Node B*. Subtracting (25) from (24) leads to

$$T_{2,i}^{(\mathrm{P})} - T_{2,i}^{(\mathrm{B})} = \theta_{offset}^{(\mathrm{BP})} + \theta_{skew}^{(\mathrm{BP})} \cdot (T_{1,i}^{(\mathrm{A})} - T_{1,1}^{(\mathrm{A})}) + d^{(\mathrm{AP})} - d^{(\mathrm{AB})} + X_i^{(\mathrm{AP})} - X_i^{(\mathrm{AB})}. \tag{26}$$

Since $d^{(\mathrm{AB})}$ and $d^{(\mathrm{AP})}$ are fixed values and $X_i^{(\mathrm{AB})}$ and $X_i^{(\mathrm{AP})}$ are normal distributed RVs, the noise component can be defined by $z[i] \triangleq \mu' + X_i^{(\mathrm{AP})} - X_i^{(\mathrm{AB})}$, where $\mu' \triangleq d^{(\mathrm{AP})} - d^{(\mathrm{AB})}$ and $z[i] \sim \mathcal{N}(\mu', \sigma^2)$. Let define $x[i] \triangleq T_{2,i}^{(\mathrm{P})} - T_{2,i}^{(\mathrm{B})} - \mu'$ and $w[i] \triangleq z[i] - \mu'$, then the set of observed data can be written in matrix notation as follows:

$$\mathbf{x} = \mathbf{H}\boldsymbol{\theta} + \mathbf{w},$$

where $\mathbf{x} = [x[1]\ x[2]\ \cdots\ x[N]]^T$, $\mathbf{w} = [w[1]\ w[2]\ \cdots\ w[N]]^T$, $\boldsymbol{\theta} = [\theta_{offset}^{(\mathrm{BP})}\ \theta_{skew}^{(\mathrm{BP})}]^T$, and

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 0 & T_{1,2}^{(\mathrm{A})} - T_{1,1}^{(\mathrm{A})} & \cdots & T_{1,N}^{(\mathrm{A})} - T_{1,1}^{(\mathrm{A})} \end{bmatrix}^T.$$

Note that the noise vector $\mathbf{w}$ is normally distributed $\mathbf{w} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$, and the matrix $\mathbf{H}$ represents the observation matrix of size $N \times 2$. From [15, Theorem 3.2, p. 44], the least-squares estimator for the relative clock offset and skew is given by

$$\hat{\boldsymbol{\theta}} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{x}, \quad \mathbf{I}(\boldsymbol{\theta}) = \frac{\mathbf{H}^T\mathbf{H}}{\sigma^2},$$

where $\mathbf{I}(\boldsymbol{\theta})$ is the Fisher information matrix. After some straightforward mathematical manipulations, the joint clock offset and skew estimator is given by

$$\begin{bmatrix} \hat{\theta}_{offset}^{(\mathrm{BP})} \\ \hat{\theta}_{skew}^{(\mathrm{BP})} \end{bmatrix} = \frac{1}{N\sum_{i=1}^{N} D_i^2 - \left[\sum_{i=1}^{N} D_i\right]^2} \begin{bmatrix} \sum_{i=1}^{N} D_i^2 \sum_{i=1}^{N} x[i] - \sum_{i=1}^{N} D_i \sum_{i=1}^{N} [D_i \cdot x[i]] \\ N\sum_{i=1}^{N} [D_i \cdot x[i]] - \sum_{i=1}^{N} D_i \sum_{i=1}^{N} x[i] \end{bmatrix}, \tag{27}$$

where $D_i \triangleq T_{1,i}^{(\mathrm{A})} - T_{1,1}^{(\mathrm{A})}$. By inverting the Fisher information matrix $\mathbf{I}(\boldsymbol{\theta})$, the Cramer-Rao lower bounds (CRBs) for the relative clock offset and skew take the expressions:

$$\mathrm{var}(\hat{\theta}_{offset}^{(\mathrm{BP})}) \geq \frac{\sigma^2 \sum\limits_{i=1}^{N} D_i^2}{N \sum\limits_{i=1}^{N} D_i^2 - \left[\sum\limits_{i=1}^{N} D_i\right]^2}, \quad \mathrm{var}(\hat{\theta}_{skew}^{(\mathrm{BP})}) \geq \frac{\sigma^2 N}{N \sum\limits_{i=1}^{N} D_i^2 - \left[\sum\limits_{i=1}^{N} D_i\right]^2}. \tag{28}$$

Consequently, using the results in (27), *Node B* can be synchronized to *Node P*. Likewise, all the other nodes in the checked region in Fig. 4 can be simultaneously synchronized to the parent node *Node P* without any additional timing message transmissions, thus saving a significant amount of energy. Fig. 6 shows the mean square error (MSE) performance of the proposed clock offset and skew estimators for ROS. It can be seen that both clock offset and skew estimators are efficient and their performance is well predicted by the CRBs. Besides, as we will see in the next section, ROS does not exhibit any loss of synchronization accuracy in clock offset estimation when compared to RBS.

4.3.3 Receiver-Receiver Synchronization

This section presents the RRS approach [7] and compares its performance with the proposed ROS approach. RRS is an approach to synchronize a set of children nodes that receive the beacon messages from the common sender, a reference or parent node. Consider a parent (reference) node P and arbitrary nodes A and B, located within the communication range of the parent node in Fig. 7. As depicted in Fig. 8, assume that both *Node A* and *Node B* receive the i^{th} beacon from *Node P* at time instants $T_{2,i}^{(\mathrm{A})}$ and $T_{2,i}^{(\mathrm{B})}$ of their local clocks, respectively. Nodes A and B record the arrival time of the broadcast packet according to their own timescales and then exchange their time-stamps. Suppose $X_i^{(\mathrm{PA})}$ denotes the nondeterministic delay components (random portion of delays) and $d^{(\mathrm{PA})}$ denotes the deterministic delay component (propagation

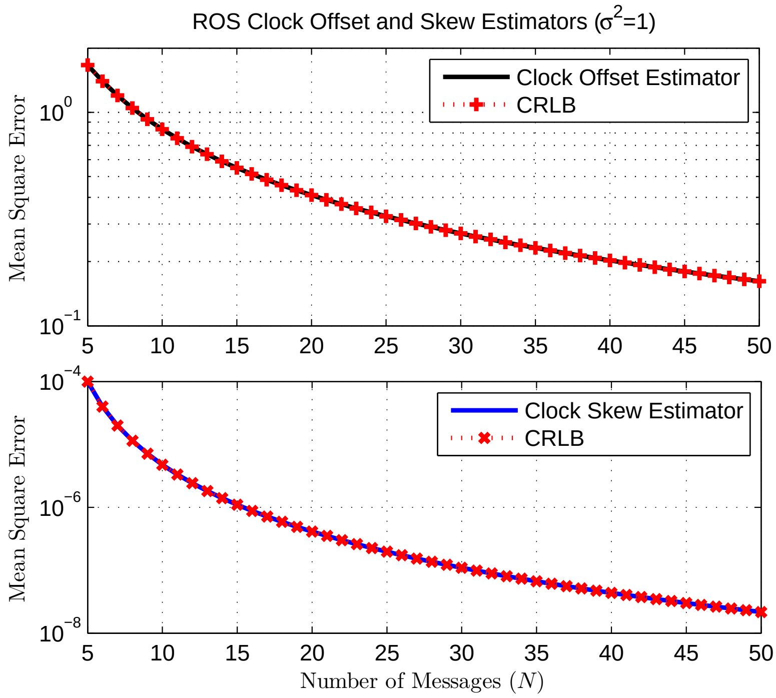

Figure 6: Performance of PBS clock offset and skew estimation.

delay) from *Node P* to *Node A*, then $T_{2,i}^{(\mathrm{A})}$ can be written as

$$T_{2,i}^{(\mathrm{A})} = T_{1,i} + d^{(\mathrm{PA})} + X_i^{(\mathrm{PA})} + \theta_{offset}^{(\mathrm{PA})} + \theta_{skew}^{(\mathrm{PA})} \cdot (T_{1,i} - T_{1,1}), \tag{29}$$

where $T_{1,i}$ is the transmission time at the reference node, $\theta_{offset}^{(\mathrm{PA})}$ and $\theta_{skew}^{(\mathrm{PA})}$ are the clock offset and skew of *Node A* with respect to the reference node, respectively. Similarly, we can decompose the arrival time at *Node B* as

$$T_{2,i}^{(\mathrm{B})} = T_{1,i} + d^{(\mathrm{PB})} + X_i^{(\mathrm{PB})} + \theta_{offset}^{(\mathrm{PB})} + \theta_{skew}^{(\mathrm{PB})} \cdot (T_{1,i} - T_{1,1}), \tag{30}$$

where $d^{(\mathrm{PB})}$, $X_i^{(\mathrm{PB})}$, $\theta_{offset}^{(\mathrm{PB})}$, and $\theta_{skew}^{(\mathrm{PB})}$ are the propagation (fixed) delay, the random portion of delays, and the clock offset and skew of *Node B*, respectively.

Subtracting (30) from (29), we obtain

$$T_{2,i}^{(\mathrm{A})} - T_{2,i}^{(\mathrm{B})} = \theta_{offset}^{(\mathrm{BA})} + \theta_{skew}^{(\mathrm{BA})} \cdot (T_{1,i} - T_{1,1}) + d^{(\mathrm{PA})} - d^{(\mathrm{PB})} + X_i^{(\mathrm{PA})} - X_i^{(\mathrm{PB})}, \tag{31}$$

where $\theta_{offset}^{(\mathrm{BA})}$ $(= \theta_{offset}^{(\mathrm{PA})} - \theta_{offset}^{(\mathrm{PB})})$ and $\theta_{skew}^{(\mathrm{BA})}$ $(= \theta_{skew}^{(\mathrm{PA})} - \theta_{skew}^{(\mathrm{PB})})$ become the relative clock offset

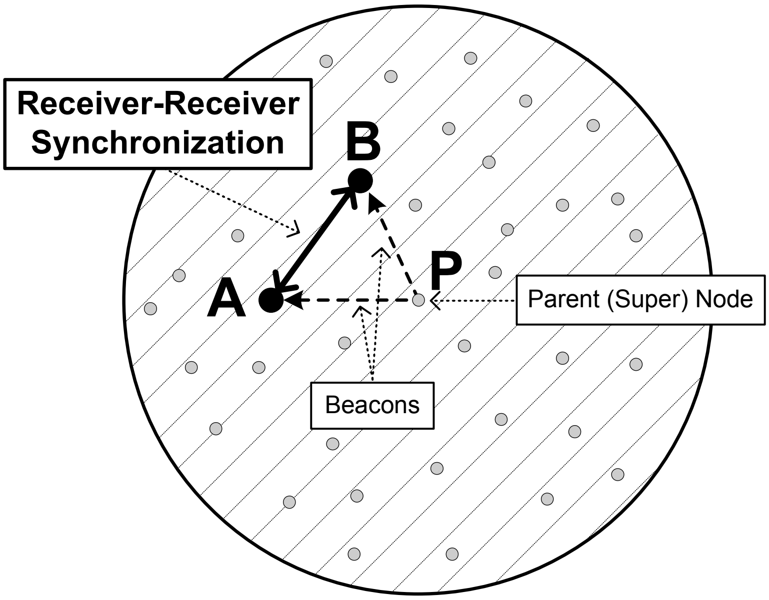

Figure 7: Receiver-receiver synchronization.

and skew between *Node A* and *Node B* at the time they receive the i^{th} broadcast packet from the reference node, respectively. Note that (31) assumes exactly the same form as (26). Hence, the same steps can be applied to derive the joint clock offset and skew estimator for ROS. Likewise, define the noise component $z[i] \triangleq \mu' + X_i^{(\mathrm{PA})} - X_i^{(\mathrm{PB})}$, where $\mu' \triangleq d^{(\mathrm{PA})} - d^{(\mathrm{PB})}$ and $z[i] \sim \mathcal{N}(\mu', \sigma^2)$. Let also define $x[i] \triangleq T_{2,i}^{(\mathrm{A})} - T_{2,i}^{(\mathrm{B})} - \mu'$ and $w[i] \triangleq z[i] - \mu'$. Using similar steps as in ROS, it is straightforward to show that the same form of the joint clock offset and skew estimator (27) can be also applied to RRS. Consequently, there is no difference between ROS and RRS with regard to the accuracy of synchronization since the effects of random delays are the same. Assuming there is no clock skew ($\theta_{skew}^{(\mathrm{BA})} = 0$), the maximum likelihood estimator of the relative clock offset $\hat{\theta}_{offset}^{(\mathrm{BA})}$ becomes

$$\hat{\theta}_{offset}^{(\mathrm{BA})} = \frac{1}{N} \sum_{i=1}^{N} \left[T_{2,i}^{(\mathrm{A})} - T_{2,i}^{(\mathrm{B})} \right], \tag{32}$$

which is the equivalent to the estimator presented in [7].

4.3.4 Comparisons and Analysis

This section compares the proposed PBS with other well-known synchronization protocols, such as TPSN, RBS, and FTSP, with respect to the amount of energy consumption (number of required timing messages) and the synchronization accuracy.

Lemma 1 *Let $N_{\mathbf{RBS}}$ be the number of required timing messages in RBS, then $N_{\mathbf{RBS}} = N + L(L-1)/2$, where L is the number of overall sensor nodes in the network.*

The reference node must broadcast the beacon packet N times in RBS. Besides, every sensor node must send time readings upon receiving the broadcast beacons with all the other nodes in the network to compensate relative clock offsets among each other [7]. Thus, $N_{\mathbf{RBS}} = N+L(L-1)/2$, since the number of unique pairs in the network is $L(L-1)/2$.

Lemma 2 *Let $N_{\mathbf{TPSN}}$ be the required number of timing messages in TPSN, then $N_{\mathbf{TPSN}} = 2N(L-1)$.*

Since every node in the network is connected to its parent node except a reference node, there are $L-1$ branches (edges) in a hierarchical tree. Besides, for TPSN, $2N$ timing messages are required in every pairwise synchronization. The number of required timing messages in TPSN is equal to the number of pairwise synchronizations times the number of required timing messages per pairwise synchronization, and therefore $N_{\mathbf{TPSN}} = 2N(L-1)$ [8].

Lemma 3 *Let $N_{\mathbf{FTSP}}$ be the number of required timing messages in FTSP, then $N_{\mathbf{FTSP}} = NL$.*

For FTSP, every sensor node must send its time readings upon receiving beacons (or broadcast beacons) to other nodes so that they can estimate the relative clock offsets among each other [22]. Therefore, the number of required timing messages in FTSP is equal to the number of sensor nodes times the number of beacons: $N_{\mathbf{FTSP}} = NL$.

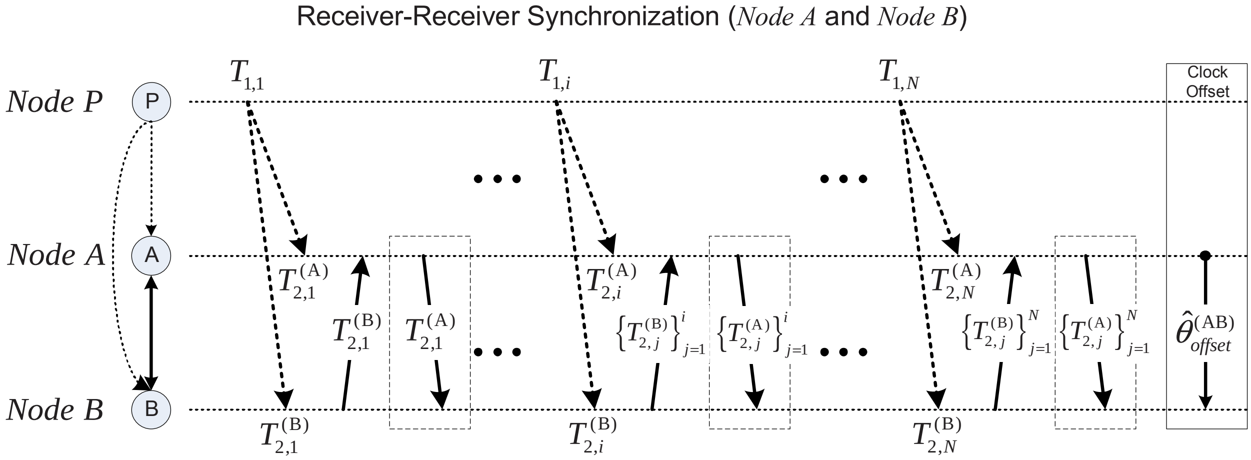

Figure 8: Clock synchronization model of RBS.

It is remarkable that the required number of timing messages for all the above mentioned protocols is proportional to the number of sensors in the network L or its square L^2. However, as discussed, PBS requires only $2N$ timing messages in every synchronization period, i.e., $N_{\mathbf{PBS}} = 2N$. Hence, $N_{\mathbf{PBS}}$ dose not depend on the number of sensors in the network, which incurs an enormous amount of energy saving. Moreover, this gain proportionally increases with respect to the scale of the network. Consequently, the benefit of PBS over RBS, TPSN, and FTSP is clear and huge in terms of energy consumption with the cost of allocating 2 super nodes in the network. Note that RBS also requires a super node which broadcasts the reference beacons to all the other nodes in the network.

In case that there are other sensor nodes which are located outside of the checked region in Fig. 4, likewise RBS, the network could be divided into a number of separated groups (clusters) and they could be synchronized by additional pairwise synchronizations among super nodes in different groups, i.e., global synchronization can be achieved by a sequence of pairwise synchronizations. Here, diverse grouping and pair selection algorithms can be considered according to the type of the network. For instance, assuming that the level hierarchy of the network is discovered by an appropriate searching algorithm (e.g., as in [8]), there exist groups of parents and children nodes, where a group consists of a parent and its children nodes. Here, every parent node can investigate the connectivity among its children nodes and select the best sequence of synchronization pairs in order to minimize the required number of pairwise synchronizations, which maximizes the number of nodes performing ROS. Note that no network-wide heuristic connection search is required in this case because of its limited and known set of scanning nodes. The detailed description of these results for the proposed PBS scheme is reported in [25] and [26].

5 Conclusions

Time synchronization is a significant component in the deployment of wireless ad-hoc networks, and a number of fundamental operations, like data fusion, power management and transmission scheduling, require accurate time synchronization. Since the conventional NTP time synchronization protocol for the Internet can not be directly applied to wireless sensor networks and wireless airborne networks, a number of synchronization protocols have been developed in this project to meet the unique requirements of these applications.

In this project, the basic features and theoretical backgrounds of the time synchronization problem in wireless sensor networks were first addressed and two general large scale network synchronization protocols were proposed. Besides, a survey of existing time synchronization protocols in the literature was provided including the most recent results. As the main contributions of this work, we can enumerate these major contributions. First, novel analytical tools and techniques to assess the performance of existing synchronization schemes were developed. Second, a family of novel joint clock offset and skew estimators based on the classical two-way message exchange model were also developed. Third, this work proposed two novel energy-efficient time synchronization protocols: Adaptive-Sync Protocol and Pairwise Broadcast Synchronization (PBS) Protocol, which require a much lesser number of timing messages (energy consumption) than existing state-of-the-art synchronization protocols, and incur no loss in synchronization accuracy.

A number of open research problems might be worth to investigate. Experimental performance evaluation and comparisons with other existing synchronization protocols represent a major open research work. More general random delay models might be needed for other network applications (such as presence of jamming, fading, node failures). Moreover, a variety of sophisticated statistical techniques, such as jackknife, bootstrap, and Gibbs sampling, could be applied to improve the performance of clock estimation. These remain open issues for another future research project.

References

[1] H. S. Abdel-Ghaffar, "Analysis of synchronization algorithm with time-out control over networks with exponentially symmetric delays," *IEEE Trans. Commun.*, vol. 50, pp. 1652-1661, Oct 2002.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks* vol. 38, no. 4, pp. 393-422, Mar. 2002.

[3] I. F. Akyildiz, D. Pompili, T. Melodia, "Underwater Acoustic Sensor Networks: Research Challenges," *Elsevier's Journal of Ad Hoc Networks,* Vol. 3, Issue 3, pp. 257-279.

[4] K. Arvind, "Probabilistic Clock Synchronization in Distributed Systems," *IEEE Transactions on Parallel and Distributed Systems*, 5(5):474487, May 1994.

[5] F. Cristian, "Probabilistic Clock Synchronization," *Distributed Computing*, 3:146158, Springer-Verlag, 1989.

[6] T. V. Dam, and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor network," *Proceedings of the ACM Conference on Embedded Network Sensor Systems (Sensys)*, 2003.

[7] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," *Proc. of the Fifth Symposium on Operating Systems Design and Implementation*, (OSDI 2002), Dec. 2002.

[8] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync Protocol for Sensor Networks," *Proc. of the First ACM Conference on Embedded Networked Sensor Systems*, (SenSys), Nov. 2003.

[9] S. Ganeriwal, D. Ganesan, H. Shim, V. Tsiatsis, and M. B. Srivastava, "Estimating Clock Uncertainty for Efficient Duty-Cycling in Sensor Networks," *ACM Sensys'05 Conference*, pp. 130-141, Nov. 2005, San Diego, CA.

[10] A. Giridhar and P. R. Kumar, "Distributed Clock Synchronization over Wireless Networks: Algorithms and Analysis," *in Proc. of the 45th IEEE Conference on Decision and Control*, pp. 4915-4920, San Diego, Dec. 2006.

[11] J. Heidemann, W. Ye, J. Wills, A. Syed, and Y. Li, "Research Challenges and Applications for Underwater Sensor Networking." In *Proceedings of the IEEE Wireless Communications and Networking Conference,* (to appear) Las Vegas, Nevada, USA, April 2006. Available online: http://www.isi.edu/∼johnh/PAPERS/Heidemann06a.html.

[12] Y. -W. Hong and A. Scaglione, "A scalable synchronization protocol for large scale sensor networks and its applications," *IEEE Journal on Selected Areas in Commun.*, vol. 23, no. 5, pp. 1085-1099, May 2005.

[13] A. Hu and S. D. Servetto, "Asymptotically optimal time synchronization in dense sensor networks," *in Proc. of the 2nd ACM International Conference on Wireless Sensor Networks and Applications (WSNA)*, pp. 1-10, Sep. 2003.

[14] D. R. Jeske, "On Maximum Likelihood Estimation of Clock Offset," *IEEE Transactions on Communications*, Vol. 53, No. 1, pp. 53-54, Jan 2005.

[15] S. Kay, Fundamentals of Statistical Signal Processing, Vol. I: Estimation Theory, vol. II: Detection Theory, Prentice-Hall, 1993.

[16] N. Khajehnouri and A. H. Sayed, "A distributed broadcasting time-synchronization scheme for wireless sensor networks," in *Proc. of ICASSP 2005*, pp. 1053-1056, Philadelphia, March 2005.

[17] J. Kong, J. Hong Cui, D. Wu, and M. Gerla, "Building underwater ad-hoc networks and sensor networks for large scale real-time aquatic applications." In *Proceedings of the IEEE MILCOM*, Atlantic City, NJ, USA, Oct. 2005.

[18] J. Lee, J. Kim and E. Serpedin, "Clock Offset Estimation in Wireless Sensor Networks Using Robust M-estimation", WASA 2008 Conference, Dallas, TX, Oct. 2008 (submitted).

[19] E. Lehman, *Theory of Point Estimation*, Wiley, 1986.

[20] A. Leon-Garcia, *Probability and Random Processes for Electrical Engineering*, 2nd ed. Reading, MA: Addison-Wesley, 1993.

[21] Q. Li and D. Rus, "Global clock synchronization in sensor networks," IEEE INFOCOM, 2004.

[22] M. Maroti, B. Kusy, G. Simon, A. Ledeczi, "The Flooding Time Synchronization Protocol". *In Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems 2004*, ACM Press, pp. 39-49.

[23] D. L. Mills, "Internet Time Synchronization: The Network Time Protocol," *IEEE Trans. on Communications*, vol. 39, no. 10, Oct. 1991, pp. 1481-1493.

[24] S. B. Moon, P. Skelly, and D. Towsley, "Estimation and removal of clock skew from network delay measurements," *Proceedings of IEEE INFOCOM*, 1, pp. 227-234, 1999.

[25] K. L. Noh, Y. C. Wu, K. Qaraqe, and B. Suter, "Extension of Pairwise Broadcast ClockSynchronization for Multi-Cluster Sensor Networks," EURASIP Journal on Advances in Signal Processing, Special Issue on Applications of Signal Processing Techniques to Wireless Sensor Networks, January 2008 issue.

[26] K. L. Noh, Y. C. Wu, K. Qaraqe, and E. Serpedin, "Time Synchronization for Wireless Sensor Networks," book chapter in the Handbook: Adaptive Processing in Wireless Communications, CRC Press, (to be published in 2008).

[27] K. Noh, Q. Chaudhari, E. Serpedin, and B. Suter, "Maximum Likelihood and Cramer-Rao Lower Bound for Clock Offset and Skew Estimation in Timing Synch Protocol for Sensor Networks," *IEEE Trans. on Communications,* vol. 55, no. 4, pp. 766-777, Apr. 2007.

[28] K. Romer, "Time Synchronization in Ad Hoc Networks," *Proceedings of ACM Symposium on Mobile Ad Hoc Networking and Computing*, pp. 173182, Oct 2001.

[29] B. M. Sadler, "Local and broadcast clock synchronization in a sensor node," *IEEE Signal Processing Letters*, vol. 13, no. 1, pp. 9-12, Jan. 2006.

[30] B. M. Sadler and A. Swami, "Synchronization in Sensor Networks: an Overview," in *Proc. of Military Communications Conference 2006*, Washington D.C., Oct. 2006.

[31] I. Sari, K. L. Noh, Q. Chaudhari, E. Serpedin, and B. Suter, "On the Joint Synchronization of Clock Offset and Skew in RBS-Protocol," *IEEE Trans. on Communications*, vol. 56, no. 5, May 2008.

[32] M. L. Sichitiu and C. Veerarittiphan, "Simple, accurate time synchronization for wireless sensor networks," WCNC 2003.

[33] W. Su, and I. F. Akyildiz, "Time-diffusion synchronization protocol for wireless sensor networks," *IEEE/ACM Transactions on Networking*, Volume 13, No. 2, pp. 384- 397, Apr 2005.

[34] B. Sundararaman *et al.*, "Clock synchronization for wireless sensor networks: a survey," *Ad-Hoc Networks*, vol. 3, no. 3, pp. 281-323, Mar. 2005.

[35] A. Syed and J. Heidemann, "Time Synchronization for High Latency Acoustic Networks," Technical Report ISI-TR-2005-602, USC/Information Sciences Institute, April 2005. Available online: http://www.isi.edu/∼johnh/PAPERS/Heidemann06a.html.

[36] Z. Tian, X. Luo, and G. B. Giannakis, "Cross-layer sensor network synchronization," in *Proc. of the 38th Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 1276-1280, Nov. 2004.

[37] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," *Proceedings of the 21st International Conference of the IEEE Computer and Communications Societies (Infocom)*, 2002.

6 List of Symbols, Abbreviations, and Acronyms

AO = Always On

Adaptive Sync = Adaptive Synchronization

CRLB = Cramer-Rao Lower Bound

FTSP = Flooding Time Synchronization Protocol

GPS = Global Positioning System

ID = Identification

iid = independent and identically distributed

IP = Internet Protocol

MAC = Medium Access Control

ML = Maximum Likelihood

MLE = Maximum Likelihood Estimator

MSc = Master of Science

MSE = Mean Square Error

NTP = Network Time Protocol

PBS = Pairwise Broadcast Synchronization

PDF = Probability Density Function

PhD = Doctor of Philosophy

RBS = Reference Broadcast Synchronization

RF = Radio Frequency

ROS = Receiver Only Synchronization

RRS = Receiver Receiver Synchronization

RV = Random Variable

SI = Sensor Initiated

SRS = Sender Receiver Synchronization

TDMA = Time Division Multiple Access

TDP = Time Diffusion Protocol

TPSN = Time Protocol for Synchronization of Sensor Networks

UAV = Unmanned Aerial Vehicles

UDP = Universal Data Protocol

WAN = Wireless Airborne Network

WLAN = Wireless Local Area Network

wrt = with respect to

WSN = Wireless Sensor Network

7 List of Publications Resulted from this Project

[1] K. L. Noh, Q. Chaudhari, E. Serpedin, and B. Suter, "Novel Clock Phase Offset and Skew Estimation Using Two-Way Timing Message Exchanges for Wireless Sensor Networks," IEEE Transactions on Communications, vol. 55, no. 4, April 2007, pp. 766-777.

[2] I. Sari, K. L. Noh, Q. Chaudhari, E. Serpedin, and B. Suter, "On the Joint Synchronization of Clock Offset and Skew in RBS-Protocol," IEEE Trans. On Communications, vol. 56, no. 5, May 2008.

[3] Q. Chaudhari, E. Serpedin, and K. Qaraqe, "On Maximum Likelihood Estimation of Clock Offset and Skew in Wireless Sensor Networks with Exponential Delays," IEEE Trans. On Signal Processing, . vol. 56, no. 4, April 2008.

[4] K. L. Noh, Y. C. Wu, K. Qaraqe, and B. Suter, "Extension of Pairwise Broadcast ClockSynchronization for Multi-Cluster Sensor Networks," EURASIP Journal on Advances in Signal Processing, Special Issue on Applications of Signal Processing Techniques to Wireless Sensor Networks, January 2008.

[5] Q. Chaudhari and E. Serpedin, "A simple algorithm for clock synchronization in wireless sensor networks," IEEE International Workshop "From Theory to Practice in Wireless Sensor Networks," (T2PWSN 2007), Helsinki, Finland, June 2007. This workshop was part of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks.

[6] K. L. Noh, and E. Serpedin, "Pairwise broadcast synchronization in wireless sensor networks," IEEE International Workshop "From Theory to Practice in Wireless Sensor Networks," (T2PWSN 2007), Helsinki, Finland, June 2007. This workshop was part of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks.

[7] Q. Chaudhari, Y. Zhou, and E. Serpedin, "Efficient Clock Synchronization in Wireless Sensor

Networks," Asilomar Conference 2007, Pacific Grove, CA, June 2007 (submitted for publication).

[8] K. L. Noh, Y. C. Wu, K. Qaraqe, and E. Serpedin, " Time Synchronization for Wireless Sensor Networks," book chapter in the Handbook: Adaptive Processing in Wireless Communications, CRC Press, Submitted July 2007, (to be published in 2008).

[9] Q. Chaudhari and E. Serpedin, "On maximum likelihood estimation of clock phase, skew and drift in networks with exponentially distributed delays," EURASIP Journal on Advances in Signal Processing (accepted), Volume 2008, Article ID 219458, 6 pages

[10] J. Lee, J.S. Kim, and E. Serpedin, "Robust bootstrap based clock offset estimators for wireless sensor networks," WASA 2008 The International Conference on Wireless Algorithms, Systems and Applications (WASA'08), October 26-28, 2008, Dallas, (submitted).

[11] Q. Chaudhari and E. Serpedin, "Clock Offset and Skew Estimation in Wireless Sensor Networks with Known Deterministic Delays and Exponential Nondeterministic Delays," The Third International Conference on Digital Telecommunications (ICDT 2008), June 29 - July 5, 2008 (submitted).

[12] Q. Chaudhari, E. Serpedin, and K. Qaraqe, "Statistical Signal Processing for Clock Synchronization in Wireless Sensor Networks," WoSPA 2008 Conference

[13] Q. Chaudhari and E. Serpedin, "Energy efficient synchronization algorithms for wireless sensor networks," EUSIPCO 2008 Conference, (submitted).

[14] Q. Chaudhari and E. Serpedin, "Clock Drift Estimation in Networks with Symmetric Exponential Link Delays," submitted to 4th Advanced International Conference on Telecommunications (AICT), June 2008.

[15] Kyoung-Lae Noh, "NEW ADVANCES IN DESIGNING ENERGY EFFICIENT TIME SYNCHRONIZATION SCHEMES FOR WIRELESS SENSOR NETWORKS," PhD Thesis Dissertation, Texas A&M University, Dept. of Electrical and Computer Engineering, College Station,

TX, August 2007.